Week 9: Effort Estimation & COCOMO

Rachelle Bosua

rachelle.bosua@unimelb.edu.au

www.lms.unimelb.edu.au

Outline

- Continue with COCOMO
  - Basic principles
  - Steps
  - Tools
- AGILE & Project Management

Project Planning with a focus on estimating

1) Software Scoping
2) Estimate the resources required for SW dev
3) Software Project Estimation
4) Decomposition techniques for estimation
5) Empirical Estimation (COCOMO)
6) Make or Buy Decisions

5) Empirical Estimation (COCOMO)

5) COCOMO

- **CO**nstructive **CO**st **M**odel (Barry Boehm - 1981)
- Model 1: Basic

  SW dev effort & Cost – LOC
- Model 2: Intermediate

  SW dev effort & Cost =

  f (program size and set 'cost' drivers)
- Model 3: Advanced

  Intermediate with cost driver impact on each SE process step (analysis, design etc.)

5) COCOMO II

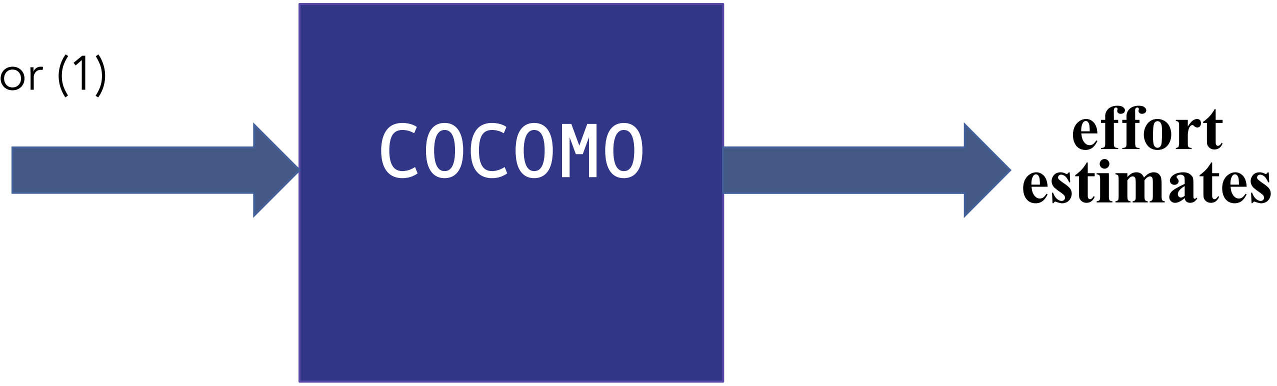
- Three different models:
    1) Application composition model
    2) Early design stage model
    3) Post-architectural stage model

Size estimates required

No Application Points for (1)

No early design FPs (2)

No FPs/LOC for (3)



COCOMO

effort estimates

1) COCOMO II Application Composition Model

Basic steps of COCOMO 2

1. Identify the basic Application points

- Application points is a weighted measure of: # of screens displayed in the application, # of reports produced and # of program components likely to be developed

2. Classify the complexity of the Application points

- Categories of application (screen, report and component) are each classified as either SIMPLE, MED or DIFFICULT (similar to FPA and Table 7.1)
- For Application points rows: # views in a screen & # sections in a report. Columns =# data tables on client server machines (for screens and report)

3. Calculate the total number of Application points

- Calculate in similar way as for total # of function points. Multiply the count of application points by weight of type and sum over all these for total application point count

Formulae

$$\sum_{i=1}^{3}(AT_i \times W_i),$$

Application Type	Complexity Weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
Component	-	-	10

Complexity weights for application types in COCOMO

$$E = \frac{NAP}{PROD}.$$

Basic steps of COCOMO 2

4. Estimate the productivity rate

- The number of points a team can do per person months. This estimate depends on the experience of the development team for the particular type of application, the maturity and capability of the development environment

Category	Description	
	Developer	Environment
Very low	< 5 months experience	Edit, Code, Debug
Low	> 5mnts, < 9mnts	Simple front end, with back end, little integration
Nominal	> 9mnths, < 1yr experience	Basic lifecycle tools, moderately integrated
High	> 1yr, < 2yrs exp	Strong, mature lifecycle models moderately, integrated
Very high	> 2 yrs exp.	Strong, mature, pro-active lifecycle tools, well integrated processes, methods and reuse

Basic steps of COCOMO 2

Dev’s experience and capability	V Low	Low	Nominal	High	V High
Environmnt maturity & capability	**V Low**	**Low**	**Nominal**	**High**	**V High**
PRODUCTIVITY	4	7	13	25	50

Table to estimate the productivity rate of a project

5.
Calculate the total effort

- Calculate the total effort estimate of the project. If NAP is the number of application points, and PROD is the productivity, then the total effort is:

$$NAP_{new} = NAP \times \frac{100 - r}{100}.$$

2) COCOMO II Early Design and Post-architectural Models

COCOMO 2-Early design

More sophisticated (& accurate) models – using parameters & coefficients

$$E = bS^c m(\overrightarrow{X})$$

bS^c = initial size estimate adjusted by **m(X)**

value of b=2.94

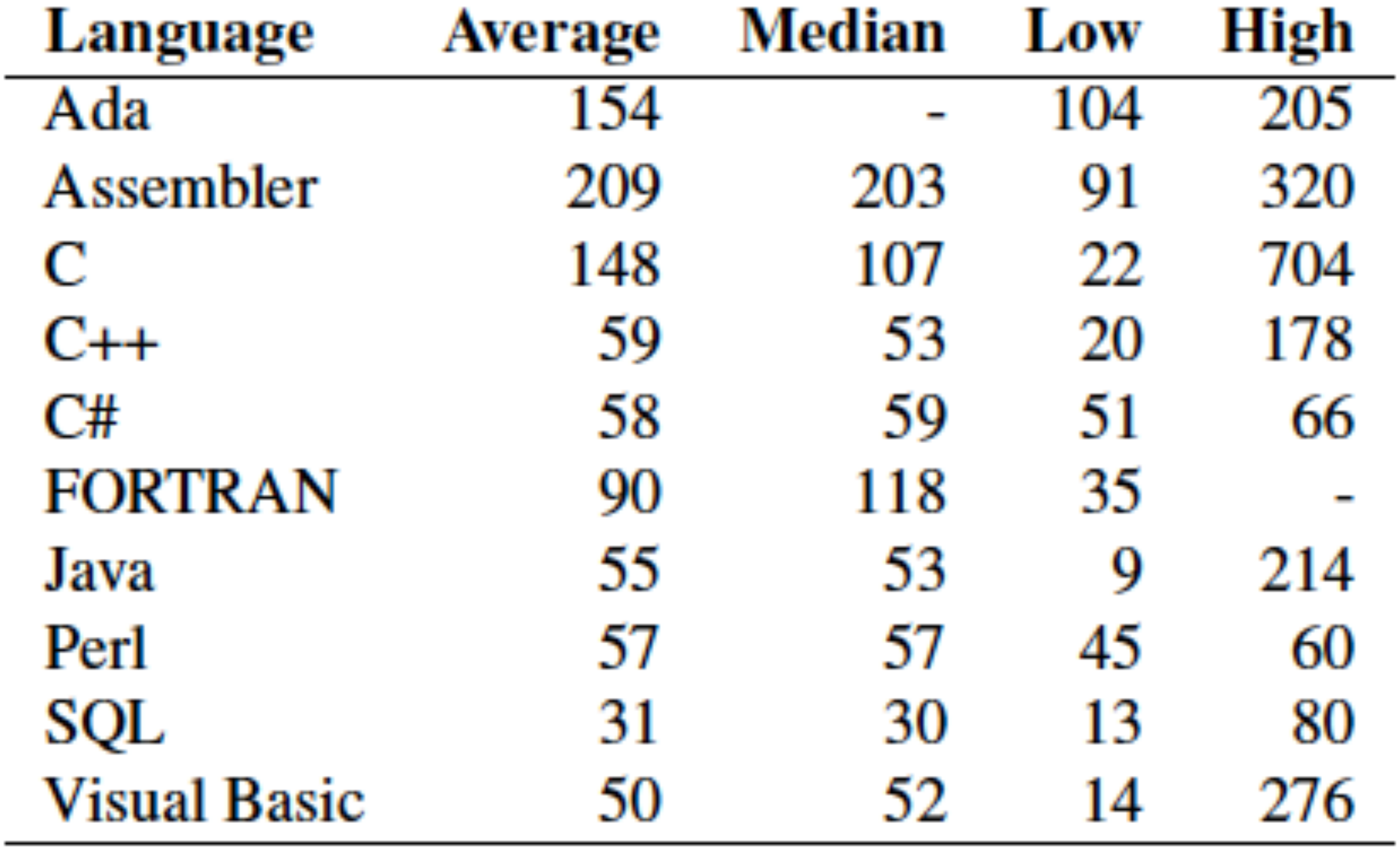
value of c based on **5 Scaling factors:**

1) Estimation size
2) Estimation scale
3) Estimating cost driver influence
4) Calculation of the effort and
5) Calculation of time and personnel

1) Estimating size

- S most influential in estimation

No logical lines of code / FP for programming lang’s



Language	Average	Median	Low	High
Ada	154	-	104	205
Assembler	209	203	91	320
C	148	107	22	704
C++	59	53	20	178
C#	58	59	51	66
FORTRAN	90	118	35	-
Java	55	53	9	214
Perl	57	57	45	60
SQL	31	30	13	80
Visual Basic	50	52	14	276

2) Estimating scale

- c is calculated by the project team

$$c = 1.01 + 0.01 \sum_{i=0}^{5} W_i,$$

w^{i} one of 5 scaling factors in range 0 to 5

So the value of c can range 1.01 to 1.26

1) Precedentness
2) Development flexibility
3) Architecture completed & risks eliminated
4) Team cohesion and
5) Process maturity

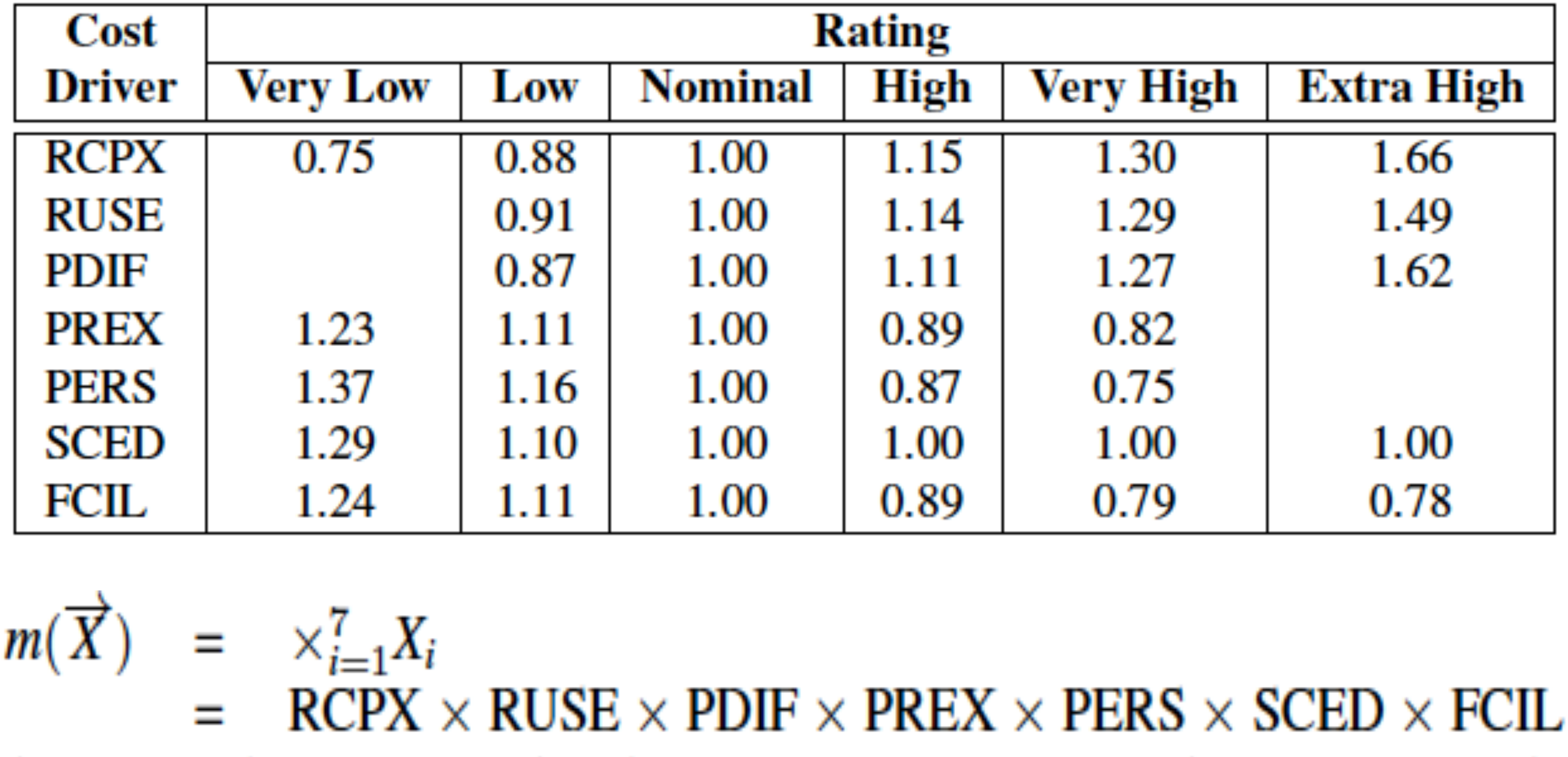
3) Est. cost/driver influence

- Estimate $m\ \overrightarrow{(X)}$ (cost drivers) [= 7]

RCPX	Expected complexity of the internal process, and level of reliability required for the system
RUSE	Level of reusability required – the level of reuse this system expected to offer to another system
PDIF	Level of platform difficulty – constraints placed on the system by the platform on which it runs, e.g. processor time and storage
PREX	Experience of the personnel on project, (<2 mnths to 6 years)
PERS	Capability of personnel on project , 15th percentil to 90th percentile
SCED	Constraints placed on project schedule - = % stretch out of schedule. Low is 75% the length of nominal project, very high is 160% of nominal
FCIL	Team support facilities

3) Est. cost/driver influence

Cost drivers for COCOMO early design model



Cost Driver	Rating					
	Very Low	Low	Nominal	High	Very High	Extra High
RCPX	0.75	0.88	1.00	1.15	1.30	1.66
RUSE		0.91	1.00	1.14	1.29	1.49
PDIF		0.87	1.00	1.11	1.27	1.62
PREX	1.23	1.11	1.00	0.89	0.82	
PERS	1.37	1.16	1.00	0.87	0.75	
SCED	1.29	1.10	1.00	1.00	1.00	1.00
FCIL	1.24	1.11	1.00	0.89	0.79	0.78

$$
\begin{aligned}
m(\overrightarrow{X}) &= \times_{i=1}^{7} X_i \\
&= \mathrm{RCPX} \times \mathrm{RUSE} \times \mathrm{PDIF} \times \mathrm{PREX} \times \mathrm{PERS} \times \mathrm{SCED} \times \mathrm{FCIL}
\end{aligned}
$$

4) Calculate the effort

$$E = bS^{c}m(\overrightarrow{X}),$$

5) Calculate the time and personnel

$$T = 2.5E^{(0.33+0.2\times(c-1.01))} \times \frac{\text{SCED(Percentage)}}{100},$$

T = estimate of nominal delivery time for a project

$$N = \frac{E}{T}$$

Worked example COCOMO II in action in CH 7 Page 93-94

Example of Online tool:
http://csse.usc.edu/tools/cocomoii.php

Estimation in an Agile world.....

COCOMO - value

- Parametric model measure proj size as complexity of what needs to be done
- Abstracts away from the SDLC
- Whatever SDLC followed, would have minimal impact

Q – do you think Software Estimation approaches such as COCOMO are valid for Agile ?

Week 8 Lecture 2 . . .

References:

Function Point:

http://yunus.hun.edu.tr/~sencer/size.html

COCOMO:

http://www.softstarsystems.com/cocomo2.htm